



Network Connections

Link submit:

https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=734

Solution:

| | |
|--------|---|
| C++ | http://ideone.com/6Vv9XL |
| Java | https://ideone.com/xuC9t0 |
| Python | https://ideone.com/HhldVf |

Tóm tắt đề:

Bob là người quản trị mạng máy tính, các máy tính được kết nối 2 chiều với nhau. Anh ấy có danh sách những câu truy vấn cho hệ thống máy tính của mình như sau:

- Nếu truy vấn là "c" thì 2 máy tính này sẽ được kết nối với nhau.
- Nếu truy vấn là "q" thì đây là câu hỏi, hỏi xem 2 máy tính có kết nối với nhau không.

Sau khi trải qua toàn bộ các câu truy vấn bạn sẽ tổng hợp lại những câu trả lời của truy vấn "q" có bao nhiêu máy tính kết nối thành công và có bao nhiêu máy tính kết nối không thành công.

Input

Dòng đầu gồm một số nguyên dương T là số lượng bộ test:

T bộ test sau, mỗi bộ test được tổ chức theo dạng như sau (mỗi bộ test cách nhau 1 dòng):

- Dòng 1 là số lượng máy tính có trong hệ thống của Bob. Số nguyên dương kiểu integer.
- Các dòng sau mỗi dòng là một truy vấn dạng "c" và "q" (bạn sẽ không biết có bao nhiêu dòng).
- Mỗi truy vấn có 3 tham số, tham số đầu tiên là "c" hoặc "q" tham số thứ 2 và thứ 3 đại diện cho computer_i computer_j.

Output

In ra T dòng ứng với T bộ test, định dạng như sau "số máy tính kết nối thành công, số máy tính kết nối không thành công. Mỗi câu trả lời cách nhau 2 dòng xuống hàng.

| | |
|---|-----|
| 2 | 1,2 |
|---|-----|

| | |
|---|-----|
| 10 c 1 5 c 2 7 q 7 1 c 3 9 q 9 6 c 2 5 q 7 5 1 q 1 1 c 1 1 q 1 1 | 2,0 |
|---|-----|

Giải thích: Trong ví dụ 1 có 3 truy vấn “q” và ta tìm được 1 kết nối thành công và 2 kết nối không thành công vì thế in ra 1, 2. Tương tự trong ví dụ 2 thì có 2 kết nối thành công và 0 kết nối không thành công in ra 2, 0.

Hướng dẫn giải:

Bài này bạn sẽ gặp khó khăn 1 chút ở việc đọc dữ liệu đầu vào, do số lượng truy vấn không cho biết trước số lượng mà chỉ cho biết trước số lượng máy tính có trong hệ thống mạng.

Nếu dùng C/C++ bạn cần đọc mỗi dòng bằng `while(getline(cin, string))` khi việc đọc vào chuỗi với kích thước chuỗi bằng 0, nghĩa là đã hết câu truy vấn.

Khi đọc xong vào string để tách từng tham số ra bạn dùng stringstream trong thư viện sstream để lấy ra từng tham số (cách nhau bởi dấu cách).

Sau đã xử lý xong phần đọc dữ liệu bạn sẽ tạo ra một mảng với độ lớn bằng độ lớn của máy tính có trong hệ thống mạng. Nếu gặp truy vấn là “c” bạn sẽ dùng hàm Unionset để kết nối 2 máy tính này với nhau.

Nếu gặp truy vấn là q thì kiểm tra xem nút cha của mỗi nút là ai nếu trùng nhau thì 2 máy tính được kết nối với nhau, ngược lại thì không có kết nối. Lưu hai biến đếm này lại và khi hết truy vấn sẽ xuất kết quả ra đúng định dạng đề bài.

Độ phức tạp: $O(T*N*Q)$ với T là số lượng bộ test, N là số lượng máy tính và Q là số truy vấn trong trường hợp sử dụng DSU cơ bản, nếu sử dụng DSU nâng cao (update by rank, path compression, by size) thì sẽ là $O(T*Q*\log(N))$